Detection of AIBO and Humanoid Robots using Cascades of Boosted Classifiers*

Matías Arenas, Javier Ruiz-del-Solar and Rodrigo Verschae

Department of Electrical Engineering, Universidad de Chile
{marenas,jruizd, rverscha}@ing.uchile.cl

Abstract. In the present article a framework for the robust detection of mobile robots using nested cascades of boosted classifiers is proposed. The boosted classifiers are trained using Adaboost and domain-partitioning weak hypothesis. The most interesting aspect of this framework is its capability of building robot detection systems with high accuracy in dynamical environments (RoboCup scenario), which achieve, at the same time, high processing and training speed. Using the proposed framework we have built robust AIBO and humanoid robot detectors, which are analyzed and evaluated using real-world video sequences.

1. Introduction

In robot soccer scenarios, the detection of teammates and opponent robots is a key skill for good playing (e.g. passing, robot avoidance, goal kicking). However, most existing systems are not robust enough in the detection of other players, mainly because they are based on pure color analysis, which is very dependent on the illumination. To revert this, we have adapted our previously developed framework for face analysis system [8] to the task of building fast robot detector systems. This framework uses nested cascades of classifiers [10], the Adaboost boosting algorithm [6], and domain-partitioning based classifiers [6]. To our knowledge these statistical learning techniques have not been used before in robot detection applications.

Using the proposed framework we have built three AIBO robot detectors (ERS7 model), each one tuned for a different pose (frontal, profile and back), and also a humanoid robot detector. The main strengths of the developed robot detection systems are: the ability of working at multiple scales, being illumination invariant to a larger degree (they work in grey scale images and no preprocessing is needed for photometric normalization), and being near real-time.

The article is structured as follows. In section 2 some related work is outlined. In section 3 the robot detection framework is described. The training procedures for building AIBO and Humanoid robot detectors are described in section 4. In section 5 an evaluation of the developed robot detectors is presented. Finally, in section 6, some conclusions of this work are given.

* This research was partially supported by FONDECYT (Chile) under Project Number 1061158.

2. Related Work

Several approaches have been proposed to tackle the object detection problem. In the case of the RoboCup competition, most approaches for detecting robots are based on pure color segmentation and on the detection of contrast changes using scan lines (see for example [3][4]). These simple approaches are not robust enough; they are highly dependent on the illumination and background. In [2] is proposed a detection system for AIBO robots based on the use of local image descriptors and SIFT features, but its main limitations are its low processing speed and its reduced performance when highlights are present in the image, which are common in AIBO robots. However, if we consider other object detection problems, there are many robust approaches that are based on statistical classifiers [1], including systems based on neural networks, PCA projections, decision trees, SVM classifiers, and cascades of boosted classifiers.

Generally, one of the main drawbacks of detection systems based on statistical classifiers is that they are not real-time. The systems based on cascades of boosted classifiers, however, are an exception; they are very fast and accurate at the same time. The Viola&Jones classifier [9] use a cascade of filters for a fast classification, where each filter is trained using Adaboost, and the integral image for fast computation of the features, which are based on simple, rectangular features (a kind of Haar wavelets). This kind of classifier allows obtaining fast processing speed and high detection rates. These ideas are further improved in [10], where *nested* cascades are introduced. Nested cascades reuse the confidence output of a given layer, in the next layer of the cascade, which allows obtaining more compact (faster) cascades and more accurate classifications. It also uses domain-partitioning weak classifiers [6], which, compared to [9], achieves an improvement in the representation power of the weak classifiers and reduces the processing and training time. In [8] a procedure to train nested cascades of boosted classifiers that allows to considerably reduce the training time (from months in [9] to a few days) is proposed. A second improvement proposed in [8] is the use of both internal and external bootstrap for the training of the cascade. A third improvement corresponds to a criterion to automatically select the number of weak classifiers in each layer of the cascades, which aims to minimize the processing time and at the same time assures a high detection rate and a very low false positive rate. This learning framework [8] has been extended in this work to the task of robot detection.

3. Robot Detection Framework

We briefly describe the developed multiscale robot detection framework (see block diagram in figure 1). First, to detect the robots at different scales, a multiresolution analysis of the images is performed, by downscaling the input image by a fixed scaling factor --e.g. 1.2-- (*Multiresolution Analysis* module). This scaling is performed until images of about 24x24 pixels are obtained. Afterwards, windows of 24x24 pixels are extracted in the *Window Extraction* module for each of the scaled versions of the input image. The extracted windows could then be pre-processed to obtain invariance against changing illumination, but thanks to the used of illumination invariant features we do not perform any kind of preprocessing. Afterwards, the windows are

analyzed by a nested cascade classifier (*Cascade Classification Module*). Finally, in the *Overlapping Detection Processing* module, the windows classified as positive (they contain a robot) are fused (normally a robot will be detected at different scales and positions) to obtain the size and position of the final detections.

Using the described framework it is also possible to detect the robots pose. To achieve this, detectors tuned to different robot poses/views (e.g. frontal, profile and back) should be trained and applied. In general terms there are two possible forms of applying the detectors. The first one consists in applying the detectors in parallel. Then, the robot pose will be given by the detector having the largest confidence value. The second form consists in applying first a generic detector (not tuned to any pose) and then, in the *pose classification module,* verifying the detection, and also obtaining the pose of the robot applying the pose-specific detectors in parallel.

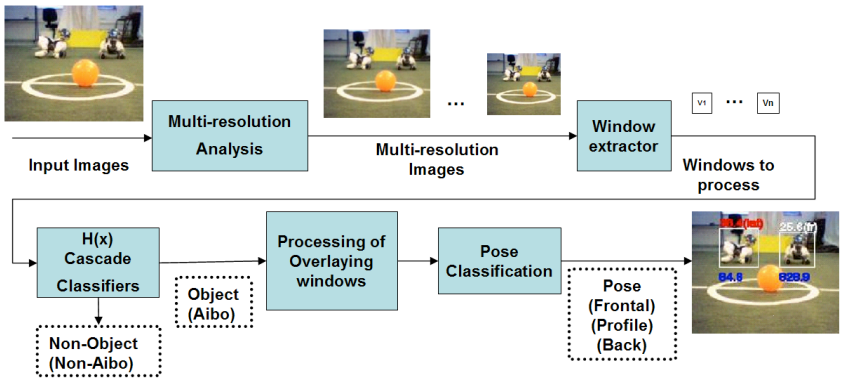

Figure 1. Block diagram of the detection system.

3.1 Learning using Nested Cascades of Classifiers

A nested cascade of boosted classifiers is composed by integrated layers, each one containing a boosted classifier. The cascade works as a single classifier that integrates the classifiers of every layer H_C^k , defined as:

$$H_C^k(x) = H_C^{k-1}(x) + \sum_{t=1}^{T_k} h_t^k(x) - b_k \qquad (1)$$

with $H_C^0(x) = 0$, h_t^k the weak classifiers, T_k the number of weak classifiers in layer k, and b_k a threshold (bias) value that defines the operation point of the strong classifier. The class assigned to the output corresponds to the sign of $H(x)$. The output of H_C^k is a real value that corresponds to the confidence of the classifier, and its computation makes use of the already evaluated confidence value of the previous layers. For details on the handling of the tradeoff between the speed and the accuracy of the cascade classifier see [8].

Domain-partitioning weak hypotheses make their predictions based on a partitioning of the domain X into disjoint blocks $X_1,\ldots,X_n$, which cover all X, and for which $h(x)=h(x')$ for all $x, x' \in X_j$. Thus, the weak classifiers prediction depends only on which block X_j a given sample instance falls into. Herein the weak classifiers are applied over features, with each feature domain F being partitioned into disjoint

blocks $F_1, \ldots, F_n$, and a weak classifier h having an output for each partition block of its associated feature f: $h(f(x)) = c_j \ni f(x) \in F_j$

For each classifier, the value associated to each partition block (c_j) is set to minimize a loss function on the margin [6]. This value depends on the number of times that the corresponding feature, computed on the training samples (x_i), fall into this partition block (histograms), and on the class of these samples (y_i) and their weight $D(i)$:

$$c_j = \frac{1}{2}\ln\left(\frac{W_{+1}^j + \varepsilon}{W_{-1}^j + \varepsilon}\right), \quad W_l^j = \sum_{i: f(x_i) \in F_j \wedge y_i = l} D(i) = \Pr\left[f(x_i) \in F_j \wedge y_i = l\right] \text{where } l = \pm 1 \qquad (2)$$

where ε is a regularization parameter. The outputs, c_j, from each of the weak classifiers, obtained during training, are stored in a LUT to speed up its evaluation. The Adaboost learning algorithm is employed to select the features and the weak classifiers $h_t^k(x)$. We use simple, rectangular features (a kind of Haar wavelets) [9].

3.2. Selection of the training examples

Every window of any size in any image that does not contain an object (e.g. an AIBO robot) is a valid non-object training example. Obviously, to include all possible non-object patterns in the training database is not an alternative, therefore non-object patterns that look similar to the object are selected using the bootstrap procedure [7]. This procedure corresponds to iteratively train the classifier, each time adding to the negative training set, negative examples that were incorrectly classified. According to our experience, it is important to use bootstrap in both situations: before starting the training of a new layer and for re-training a layer that was just trained. The *external* bootstrap is applied just one time for each layer, before starting its training, while the *internal* bootstrap can be applied several times during the training of the layer. The bootstrap procedure in both cases is the same with only one difference, before starting an external bootstrap all negative samples collected for the training of the previous layer are discarded (see [8] for details).

4. Training of the AIBO and Humanoid Robot Detectors

During the training of the cascades, validation and training sets are used. The procedure to obtain both sets is analogous, so only the training dataset is explained. To obtain the training set used at each layer of the cascade classifier, two types of databases are needed: one of cropped windows of positive examples (e.g frontal AIBOs) and one of images not containing the object to be detected. The second type of database is used during the bootstrap procedure to obtain the negative examples. The training dataset is used to train the weak classifiers, and the validation database is used to decide when to stop the training of a layer and to select the bias values of the layer. To obtain positive examples (cropped windows) a rectangle bounding the robot was annotated and a square of size equal to the largest size of the rectangle was cropped and downscaled to 24x24 pixels. In the case of the humanoid robots, two windows were cropped from each robot used during training, one corresponding to the upper half of the robot (torso and head) and the other to the lower part (mostly legs).

This was made to allow the detection of either the upper or the lower part of the robot independently (using only one detector). This information should be sufficient for a successful detection under partial occlusions.

In the case of the databases used to train the AIBO detectors, the positive examples were obtained from videos captured using the AIBOs cameras and using external cameras. The videos were acquired under real-world playing conditions (variable illumination, occlusions, etc.). The sources used to build the humanoids training and validation sets were videos obtained using the same camera employed in our humanoid robots (Philips ToUCam III - SPC900NC), and videos from other humanoids obtained from the the RoboCup Humanoid league website (Hajime, Artisti, BreDo Brothers, DarmstadtDribbler and ToinPhoenix). The number of images used in each database is shown in Table 1.

Table 1. Summary of the databases used for training

Class	# Positive examples		# Negative images	
	(Training)	(Validation)	(Training)	(Validation)
Frontal AIBOs	3115	3115	5946	2550
Left AIBOs	4263	3624	5946	2550
Back AIBOs	1528	1528	5958	2562
Humanoids	3506	3500	5958	2562

5 Evaluation of the Detectors

The detection results are presented in terms of Detection Rate (DR) versus Number of False Positives (FP) in the form of ROC curves (Receiver Operation Characteristic curves) and tables, while the pose estimation results are presented using the confusion matrix. An analysis of the processing speed of the system is also presented. To evaluate the proposed system, two databases were used: one for the AIBOs (called AIBODetUChileEval) and one for the Humanoids (called HDetUChileEval). These databases were made available in http://vision.die.uchile.cl for future comparisons. No image of the training or the validation set are part of these databases. The AIBODetUChileEval database contains AIBOs in three poses (frontal, profile, back), while the HDetUChileEval database consists of images containing humanoids (from videos dribblers2006communication and dribblers2006Kicktrick). These images are from real world scenarios; containing changes in illumination, contrast, and background (see Table 2 for datils).

The performance of the proposed robot detection systems are presented in terms of DR versus FP (se Table 3 and Figure 2), and percentage of correct pose classification (Table 4). In figure 3 selected images with detection results are shown. In the AIBOs database, the first test consisted in evaluating each detector independently on the specific class it was trained to detect (e.g. “Frontal” detecting “Frontal” AIBOs). In this evaluation, AIBOs appearing under poses different to the ones being detected were not counted as false positives or correct detections. The best performing detector was the profile detector with a 90.7% DR and 70 FP (from all 724 images). The second test consisted in evaluating the performance of a particular detector when detecting all poses, including the ones they were not trained to detect. In this case the detectors were able to find AIBOs in all poses, showing a reasonably good detection rate; e.g.

the Frontal detector obtained a 90% DR of AIBOs under all poses with 392 FP. The third test (*Multiple detectors in all AIBOs*) consisted in running all AIBOs detectors (Frontal, Profile and Back) in parallel. Given that in some cases the three detectors detected the same AIBOs, the final detections were obtained by selecting all non-overlaying detections, and merging overlaying detections by choosing the one with highest confidence. It is important to notice that in this case the number of false positives slightly increased, e.g. a DR of 94.8% was obtained with 392 FP. In other words, it is possible to arbitrate among the output of the detectors without increasing considerably the number of FP, although it is about 3 times slower than the individual detectors. The humanoid detector also shows high detection rates. A 92.2% detection rate was obtained with 123 false positive in a total of 244 images. This is quite high considering that the system was training using examples corresponding to different humanoid robot models than the ones used in the evaluation.

The last test made was a pose classification of the AIBOs. For this, the frontal detector was used as a generic detector (using the same parameters that obtained a 90% DR 392 FP), followed by a verification of the detections using the specific detectors. Afterwards, the pose was estimated by taking the output of the specific detector that gave the largest confidence value. Out of the 912 detected AIBOs, 657 were "pose estimated", from which 519 were correctly estimated (79% correct classification rate). Table 4 shows the confusion matrix of the pose estimation for these AIBOs. The "Frontal" and "Profile" classifiers show the best results, classifying correctly 90% and 80% of the "Frontal" and "Profile" AIBOs, respectively.

Table 2. Summary of the evaluation databases.

Test database	#Images	#Frontal AIBOs	#Profile AIBOs	#Back AIBOs	#Humanoids	Image size
AIBODetUChileEval	724	344	489	180	-	208x160
HDetUChileEval	244	-	-	-	493	640x480

Table 3. Selected operation points (Detection Rate versus Number of False Positives) of the evaluated AIBO and Humanoids detectors.

Detector / Target	**DR %**	**FP**	**DR %**	**FP**	**DR %**	**FP**	**DR %**	**FP**	**DR %**	**FP**
Frontal /Frontal AIBOs			89.4	254	84.4	57			74.5	18
Profile / Profile AIBOs	94.7	98	90.4	70			81.3	42		
Back / Back AIBOs			89.9	166	85.6	76	79.8	27		
Frontal / All AIBOs			90.0	392			82.9	183	73.4	95
Multiple / All AIBOs	94.8	392	88.6	183	84.3	114	80.1	52		
Humanoids	94.8	590	92.2	123					75.9	3

Table 4. Confusion Matrix: AIBO pose estimation using the detection system.

True Class / Predicted Class	**Frontal AIBOs**	**Profile AIBOs**	**Back AIBOs**
Frontal AIBOs	91.63 %	11.64 %	33.87 %
Profile AIBOs	3.72 %	81.45 %	15.32 %
Back AIBOs	4.65 %	6.92 %	50.81 %

The processing time of the proposed detectors in the AIBO ERS7 robots was evaluated. ERS7 robots have a 64bit RISC Processor (MIPS R7000) from 576 MHz, 64MB RAM, and a color-camera of 208x160 pixels that delivers 30fps. Table 5 shows the average frame rate delivered by the "Frontal" AIBO detector in an ERS7 robot running the full four-legged Uchile1 control library [5], and in a 1.73 GHz Intel

Core Duo laptop with 1GB of RAM, running Windows XP. The frame rate depends mainly on the scaling factor, and the number scales skipped by the detection system. The detector still works fine with a scaling factor of 1.2 and skipping 1 or 2 of the first scales, which allows obtaining 6.3 fps in the AIBOs. This allows using the detector in our four-legged team, considering that it is not necessary to detect the robots in each frame, but every 3-7 for frames (every 90-210 milliseconds).

Table 5. Processing time of the frontal AIBO detector.

Configuration	Frame Rate (in fps) in Laptop PC		Frame Rate (in fps) in AIBO CPU	
	scaling 1.15	scaling 1.2	Scaling 1.15	scaling 1.2
no scale skipped	3.4	4.8	1.7	2.1
skip 1st scale	6.7	9.1	3.5	4.9
skip 1st,2nd scale	9.1	12.5	4.9	6.3
skip 1st,2nd,3rd scale	11.1	16.7	6.1	7.8

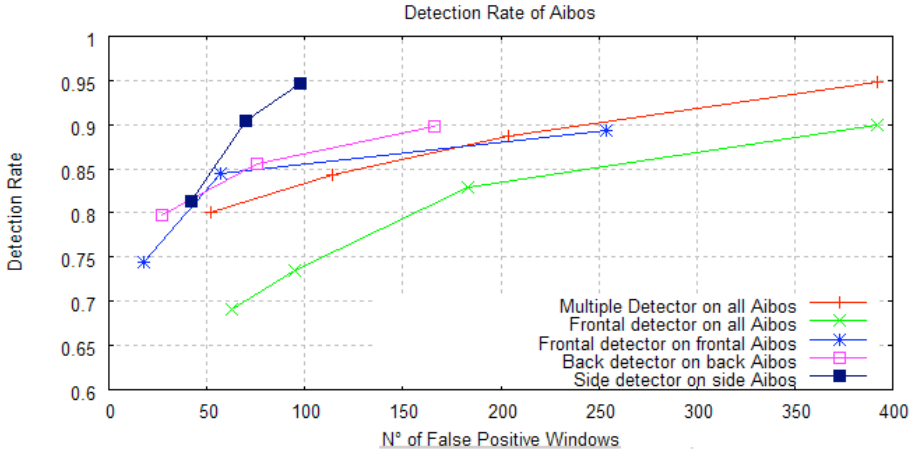

Figure 2. ROC curves (Detection Rate versus Number of False Positives) on the AIBODetUChileEval database for the Frontal, Back and Profile detectors. See text for details.

6. Conclusions

A framework for the robust detection of mobile robots using nested cascades of boosted classifiers was proposed. This framework was used to build robot detectors (Humanoids, and Frontal-, Profile- and Back-AIBOs). The main module of the system corresponds to a nested cascade of boosted classifiers, which is designed to perform fast detections with high DR and very low FPR. Using this cascade classifier, an exhaustive multiscale search is performed to be able to detect the robots appearing at different scales and positions. The detection rate of the obtained systems is quite high; for example a 90% DR with an average of 0.1 false positives per frame (208x160 pixels) is obtained for the "profile" AIBO detector, and a 92.2% DR with 123 false positives in 244 images (640x480 pixels) is obtained for the Humanoid detector. This shows that the detectors are working with high performance in difficult

environment, and still maintain good results. Even thought the detection system was not designed to estimate the pose of the AIBO robots, it was possible to estimate it with a good accuracy in the case of the AIBOs. For example, the system correctly estimated the pose in 79% percent of the detected and verified AIBOs.

The main disadvantage of the detectors is that they achieve relatively low frame rates (e.g. 6.3 fps running in the AIBO robots). Nevertheless they can be improved in several ways. First, it is not necessary to detect the robots in each frame, but every 3-7 for frames (every 90-210 milliseconds). The processing time and the number of false positives can be greatly reduced by adding the use of color-based methods and information about the location of the robot in the field (by reducing the search region area). The system can be further improved by performing a tracking of the robots.
REF

(a) (b)

Figure 3. Detection results of both detectors on the HDetUChileEval database are shown.

References

1. Hjelmås E, Low BK., "Face detection: A survey", Computer Vision and Image Understanding 83, 236-274, 2001
2. Loncomilla P., Ruiz-del-Solar J., "Gaze Direction Determination of Opponents and Teammates in Robot Soccer", LNCS 4020 (RoboCup 2005), Springer, pp. 230–242, 2006
3. Quinlan M. J. *et al.*, "The 2005 NUbots Team Report", RoboCup 2005, Four-legged league. Available on February 2006 in: http://www.robots.newcastle.edu.au/publications/NUbotFinalReport2005.pdf
4. Röfer T. *et al.*, "German Team 2005 Technical Report", RoboCup 2005, Four-legged league. Available on February 2006 in: http://www.germanteam.org/GT2005.pdf
5. Ruiz-del-Solar J., *et al.*, "UChile Kiltros 2007 Team Description Paper", RoboCup 2007 Symposium, July 9 – 10, Atlanta, USA (CD Proceedings), 2007.
6. Schapire R.E., Singer Y., "Improved Boosting Algorithms using Confidence-rated Predictions", *Machine Learning*, 37(3):297-336, 1999
7. Sung K., Poggio T., "Example-Based Learning for Viewed-Based Human Face Detecion", IEEE Trans. Pattern Anal. Mach. Intell., Vol.20, No. 1, 39-51, 1998.
8. Verschae, R., Ruiz-del-Solar, J., Correa, M. (2007). A Unified Learning Framework for object Detection and Classification using Nested Cascades of Boosted Classifiers. *Machine Vision and Applications* (in press).
9. Viola P., Jones M., "Fast and robust classification using asymmetric adaboost and a detector cascade", Advances in Neural Inform. Processing System 14. MIT Press, 2002
10. Wu B., Ai H., Huang C., Lao S., "Fast rotation invariant multi-view face detection based on real Adaboost", 6th Int. Conf. on Face and Gesture Recognition, 79–84, 2004.